Data Types

C# programlama dilinde tipler değer ve referans tipler olmak üzere ikiye ayrılır. Bir veri tipinin bellekte kaplayacağı alan önceden belliyse bu bir value type’dır.

* Value type’lar **stack**’de referance type’lar **heap**’de tutulur.
* Stack’de tutulan bir datanın heap’e aktarılmasına **boxing**, bu işlemin tam tersine ise **unboxing** denmektedir.

Example

//Boxing

int number = 1;

object data = number;

Example

//Unboxing

object data = 1;

int number = (int)data;

* Referance type’lar null değer tutabilirler. Value type’lara null değer aktarabilmek için tip adından sonra ? sembolünün kullanılması gerekir.

Example

int? number = 1;

number = null;

Yukarıdaki sözdizim yerine Nullable yapısını kullanabilirsiniz.

Example

System.Nullable<int> number = new System.Nullable<int>();

number = null;

bool

true ve false değerleri bool türündedir.

* System.Boolean , struct.
* Kapladığı alan : 1 byte.

char

Tek karakterlik veriler char türündedir.

* System.Char , struct.
* Kapladığı alan : 2 byte.
* Veriler tek tırnak içerisinde yazılmalıdır.
* char türündeki değişkenlere ASCII ve Unicode karakterler aktarılabilmektedir.

Example

//ASCII kod char türüne dönüştürülerek aktarılır.

char A = (char)65;

Console.Write(A); //A

ENTER ve BACKSPACE gibi bazı tuşları tırnak içerisinde yazmak mümkün olmadığı için bu tuşların ASCII kodlarını char türündeki değişkenlerde saklayabilirsiniz.

Example

//13, ENTER tuşunun ASCII kodudur.

char key = (char)13;

char türündeki değişkenlerde tutulan karakterlerin ASCII kodlarını Convert.ToByte() metodu ile öğrenebilirsiniz.

Example

char B = 'B';

byte ASCII = Convert.ToByte(B);

Console.Write(ASCII); //66

Unicode karakterler char değişkenlere aktarılırken aşağıdaki sözdizim kullanılmaktadır.

Example

//B karakteri aktarılır.

char letter = '\u0042';

Console.Write(letter); //B

string

Metinsel veriler string türündedir.

* System.String , class.
* Kapladığı alan : her karakter 1 byte.
* Veriler çift tırnak içerisinde yazılmalıdır.
* string veriler için uzunluk sınırlaması yoktur.
* string verilerin içerikleri index parametresi ile okunabilmektedir.

Example

string firstName = "Eren";

//Syntax : string[index] => İlgili karakter char türünde geri döner.

Console.Write(firstName[0]); //E

|  |
| --- |
| **NOT**  String sınıfının indeksleyicisi sadece get bloğuna sahiptir. index parametresi ile içeriği okuyabilir ancak değişiklik yapamazsınız. |

object

* System.Object , class.
* Her türde veri saklayabilmektedir.

byte

* System.Byte , struct.
* Kapladığı alan : 1 byte.
* 0-255 aralığındaki tamsayıları tutabilmektedir.

sbyte

* System.SByte , struct.
* Kapladığı alan : 1 byte.
* -128 ile +127 aralığındaki tamsayıları tutabilmektedir.

short

* System.Int16 , struct.
* Kapladığı alan : 2 byte.
* -32.768 ile +32.767 aralığındaki tamsayıları tutabilmektedir.

ushort

* System.UInt16 , struct.
* Kapladığı alan : 2 byte.
* 0 ile 65.535 aralığındaki tamsayıları tutabilmektedir.

int

* System.Int32 , struct.
* Kapladığı alan : 4 byte.
* -2.147.483.648 ile +2.147.483.647 aralığındaki tamsayıları tutabilmektedir.

uint

* System.UInt32 , struct.
* Kapladığı alan : 4 byte.
* 0 ile 4.294.967.295 aralığındaki tamsayıları tutabilmektedir.

long

* System.Int64 , struct.
* Kapladığı alan : 8 byte.
* -9.223.372.036.854.775.808 ile +9.223.372.036.854.775.807 aralığındaki tamsayıları tutabilmektedir.

ulong

* System.UInt64 , struct.
* Kapladığı alan : 8 byte.
* 0 ile 18.446.744.073.709.551.615 aralığındaki tamsayıları tutabilmektedir.

float

* System.Single , struct.
* Kapladığı alan : 4 byte.
* -3.402823E+38 ile +3.402823E+38, tamsayı veya virgüllü sayı.
* Virgüllü sayıların sonuna **f** veya **F** harfi eklenmelidir.
* Maksimum 7 haneyi gösterebilir. (Aşağıdaki örnekte değişkene 9 haneli bir virgüllü sayı aktarılmıştır. Çıktıda altıncı haneden sonrasının otomatikmen yuvarlandığı görülmektedir)

Example

float number = 1.23456789f;

Console.Write(number); //1.234568

double

* System.Double , struct.
* Kapladığı alan : 8 byte.
* -1.79769313486232E+308 ile +1.79769313486232E+308 , tamsayı veya virgüllü sayı.
* Virgüllü sayıların sonuna **d** veya **D** harfi eklenebilir. (Zorunluluk yoktur)
* Maksimum 15 haneyi gösterebilir.

decimal

* System.Decimal , struct.
* Kapladığı alan : 8 byte.
* -79228162514264337593543950335 ile +79228162514264337593543950335 , tamsayı veya virgüllü sayı.
* Virgüllü sayıların sonuna **m** veya **M** harfi eklenmelidir.
* Maksimum 29 haneyi gösterebilir.

Tip Dönüşümleri

* Implicit(Kapalı tip dönüşümü) : Kod yazmanıza gerek yoktur. Tip dönüşümü otomatik olarak yapılmaktadır.

Example

//küçük tipten büyük tipe dönüşüm.

byte i = 1;

int j = i;

Example

//tamsayı tipinden ondalıklı tipe dönüşüm.

int number1 = 1;

float number2 = number1;

double number3 = number1;

decimal number4 = number1;

* Explicit(Açık tip dönüşümü) : Parantezler ile yapılmaktadır.

Example

int number1 = 1;

byte number2 = (byte)number1;

Example

float number1 = 1;

int number2 = (int)number1;

* Tip dönüşümü yaparken kullanabileceğimiz metodlar :

ToString()

Convert.ToString()

Convert.ToByte()

Convert.ToSByte()

Convert.ToInt16()

Convert.ToUInt16()

Convert.ToInt32()

Convert.ToUInt32()

Convert.ToInt64()

Convert.ToUInt64()

Convert.ToBoolean()

Convert.ToChar()

Convert.ToSingle()

Convert.ToDouble()

Convert.ToDecimal()

Convert.ToDateTime()

Byte.Parse()

SByte.Parse()

Int16.Parse()

UInt16.Parse()

Int32.Parse()

UInt32.Parse()

Int64.Parse()

UInt64.Parse()

Boolean.Parse()

Char.Parse()

Single.Parse()

Double.Parse()

Decimal.Parse()

DateTime.Parse()

* İki adet tip dönüşüm kuralı :

Example

int i = 257;

byte j = (byte)i; //bilgi kaybı olur.

Console.Write(j); //1

Example

float PI = 3.14f;

int number = (int)PI; //virgülden sonrası atılır.

Console.Write(number); //3

GetType() Metodu

Bu metod bir verinin hangi tipte bilgi içerdiğini öğrenmek için kullanılır.

Example

Console.WriteLine("Julia".GetType()); //System.String

Console.WriteLine(1.GetType()); //System.Int32

* Bu metod Object sınıfı kaynaklıdır.
* Geriye System.Type türünde bilgi döndürmektedir.

Example

object data = "Julia";

System.Type dataType = data.GetType();

Console.WriteLine(dataType.Namespace); //System

Console.WriteLine(dataType.Name); //String

Console.WriteLine(dataType.BaseType); //System.Object

|  |
| --- |
| **NOT**  Value types : bool, char, byte, sbyte, short, ushort, int, uint, long, ulong, float, double, decimal, struct, enum.  Referance types : string, object, class, delegate. |